University of Mila

Institute of mathematics and Informatics

2nd year LMD computer science

Module: ASDD3

TD Series N 04

Binary Trees

Exercise 1:

Give the declaration of a binary tree of integers and write the following operations on this tree.

1) Height (a): Returns the height of the tree.
2) Nb_leaves (a): returns the number of leaves of a tree.
3) Max(a): returns the maximum of the tree a.
4) Father (a , N): returns the father of a node identified by its address.

Exercise 2 :

write the procedure that allows you to display a binary tree of integers in the case of an **inorder traversal.**

Exercise 03 (Binary search trees) :

a. Understanding
   1) Construct a binary search tree from the following keys:

      25 60 35 10 5 20 65 45 70 40 50 55 30 15

   2) Add to the obtained tree and in order, the following elements:

      22 62 64 4 8

   3) Delete, in order, the following elements from the resulting tree:

      15 70 50 35 60 25

b. Write the following recursive functions and procedures on binary search trees:
   1) Belong(x, a): which checks whether x exists in a or not.
   2) Max(a): which returns the maximum of the tree a.
   3) Display_evens (a) to display the even elements of a tree sorted in ascending order.
   4) Father (a, N): which returns the father of a node identified by its address.

Exercise 04:

A convenience store aims to implement an automated tool to manage its stock information efficiently. We propose representing this information using a binary search tree. Each node in the tree will contain details about a product, including its reference number (integer), name (string), unit price (floating-point), available quantity (float), manufacturing date, and expiration date. Products will be inserted into the tree based on their reference numbers.

1. Define the data types to represent this stock.
2. Define operations for entering and displaying product information.
3. Write the function to insert an element into this tree.
4. Write the procedure **create(a: tree, n: integer)** to create a product tree by reading information from the keyboard or from a file.

5. Write the procedure **Sell(a, Ref , Quantity)** to remove, if possible, the quantity 'Quantity' from the product that has the reference 'Ref'. The procedure must display an error message when the sale is not possible.
6. Write the procedure **Feed(L, Ref , Quantity)** to add the quantity 'Quantity' to the product that has the reference ' Ref '. If the product does not exist, the procedure must ask for the designation and price of the product and then add this product to the list.
7. Write the procedure **expiration (a, d)** to display the products that expire before a given date 'd'.
8. Redo questions 5 and 6 now considering that **a** is a simple binary tree.

Exercise 05: additional

1. Write the following operations on binary trees:
    - Positive_sum (a: tree): which returns the sum of the positive elements of the elements of tree a.
    - is_positive (a: tree): which returns *true* if tree a is non-empty and all its elements are positive, otherwise returns *false.*
2. Rewrite the previous operations on binary search trees.

Exercise 06: additional

Redo exercise 03 by writing iterative versions for the proposed functions.

Exercise 07: additional

Using a stack, write the procedure for preorder traversal of a binary tree.

Exercise 08: additional

Write the function to create a balanced binary tree from an array of integers sorted in ascending order.